CENTRE UNIVERSITAIRE - SALHI AHMED - DE NAAMA
INSTITUT DES SCIENCES & TECHNOLOGIES
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Mémoire de fin d'études

Master en informatique
Spécialité : Systèmes d'Information

Thème

Classification supervisée des textes arabes : une approche à base d'apprentissage automatique

Présenté par :

- ALIOUA Meriem

Encadreur :

- Pr. BOUCHIHA Djelloul

Année Universitaire : 2020/2021

Remerciement

C’est avec un grand plaisir qu’on réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à la réalisation et l’aboutissement à ce travail.

Tout d’abord, je remercie mon encadreur Monsieur BOUCHIHA Djelloul pour son soutient, son sérieux, sa disponibilité, ses précieux conseils et son aide tout au long de l’élaboration de ce travail.

Je remercie les membres de jury qui ont bien voulu examiner et évaluer ce mémoire.

Nous nous acquittons, enfin, volontiers d’un devoir de gratitude et de remerciements à tous nos enseignants pour la qualité de l’enseignement qu’ils ont bien voulu nous prodiguer durant nos études, afin de nous fournir une formation efficiente.

Dédicace

Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mes chers parents pour tous leurs sacrifices, leur patience, leur soutien et leur encouragement.

A mes sœurs.

A toutes la famille **ACHOURI** et la famille **ALIOUA**.

Sans oublier tous les professeurs que ce soit du primaire, du moyen, du secondaire et de l'enseignement supérieur.

A toutes personnes qui m'ont encouragé ou aidé au long de mon parcours universitaire.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible.

Que dieu leur accorde santé et prospérité.

ALIOUA Meriem

Résumé

La classification des textes est la tâche la plus fondamentale et la plus essentielle du traitement du langage naturel. La dernière décennie a vu un essor de la recherche dans ce domaine. De nombreuses méthodes, ensembles de données et mesures d'évaluation ont été proposés dans la littérature.

Dans ce mémoire, nous avons réalisé un système qui permet de classifier les textes arabes. Nous avons utilisé quatre algorithmes pour classer les textes arabes.
Mots clés: Classification des textes, texte arabe, TAL, Machine Learning

ملخص

تصنيف النصوص هو المهمة الأساسية والمهمة لمعالجة اللغة .شهد العقد الماضي طفرة في البحث في هذا المجال .تم اقتراح العديد من الطرائق والبيانات وقياسات التقييم في البحوث السابقة .

في هذه المذكرة أنشأنا نظامًا لتصنيف النصوص العربية .استخدمنا أربع خوارزميات لتصنيف النصوص العربية.

الكلمات المفتاحية : تصنيف النصوص ، النص العربي ، المعالجة التلقائية للغة ، التعلم الآلي

Abstract

Text classification is the most fundamental and essential task in natural language processing. The last decade has seen a surge of research in this field. Numerous methods, datasets, and evaluation metrics have been proposed in the literature.

In this thesis, we have created a system that classifies Arabic texts. We used four algorithms to classify Arabic texts.

Keywords: Text classification, Arabic text, NLP, Machine Learning

Table des matières

Liste des figures

Liste des tableaux

Liste des Acronymes

- ANLP : Arabic Naturel Language Processing.
- AutoML : Automated Machine Learning .
- BERT: Bidirectional Encoder Representations from Transformers.
- BOW: Bag Of Word
- CNN: Convolutional Neural Network.
- DT: Decision Tree.
- GNN: Graph Neural Network.
- LR : Régression Logistique.
- ML : Machine Learning.
- NB : Naïve Bayésienne.
- RF : Random Forest.
- SVM : Machine à Vecteurs de Support.
- TAL : Traitement Automatique de langues.
- TALA : Traitement Automatique du Langue arabe.
- TALN : Traitement Automatique du Langage Naturel.
- TF: Term Frequency.
- IDF: Inverse Document Frequency.
- TF-IDF: Term Frequency-Inverse Document Frequency.
- UML: Unified Modeling Language.

Introduction générale

La révolution de l'information bousculée par le développement à grande échelle de l'Internet a fait exploser la quantité d'informations textuelles disponibles. De même, la vulgarisation de l'informatique dans le monde, a permis de créer d'importants volumes de documents électroniques rédigés en différentes langues dont une grande partie est rédigée en langue arabe (Ahmed et Rabah, 2019).

Face à la prolifération des documents, l'utilisateur est devenu incapable de traiter ces informations d'une façon manuelle et de sélectionner l'information pertinente dans cette gigantesque base documentaire. Cette incapacité a rendu indispensable, la construction de systèmes permettant d'automatiser le processus de recherche d'information ce qui a mené à aborder des problématiques plus variées telles que la traduction automatique et la classification automatique des textes (Toussaint, 2004).

La classification de textes est une catégorisation de type supervisée qui consiste à utiliser un ensemble de documents pré-étiquètes pour pouvoir classer de nouveaux documents. Le besoin de catégoriser des textes remonte au début des années 60, mais ce n'est qu'aux années 90 que la catégorisation de textes est devenue un domaine a part entière vu sa grande sollicitation dans de nombreux applications nécessitant l'organisation de documents tels que le filtrage de document, l'organisation de documents, l'indexation documentaire, etc. (SIMON, 2011).

Le Traitement Automatique du Langage Naturel (TALN) est la discipline s'intéressant à l'automatisation du traitement de certains aspects du langage humain. Cette discipline a connu une croissance importante ces dernières années grâce aux avancées récentes en intelligence artificielle, et est maintenant appliquée dans plusieurs domaines (Billal, 2017).

Le traitement automatique de la langue arabe a suscité l'écoulement de beaucoup d'encre scientifique durant les deux dernières décennies. Le "TAL arabe" ou ANLP (Arabic Natural Language Processing, en anglais), a connu un engouement des scientifiques dans les grands laboratoires de recherche et les grandes universités à l'instar de l'université de Stanford et l'université de Pennsylvanie aux états unis, notamment après les évènements du 11 septembre 2001. Un autre facteur à cet engouement, c'est la place qu'occupe la langue arabe dans le classement mondial des langues les plus utilisées sur Internet (la quatrième place devançant le français et l'allemand par cinq positions) (Noureddine, 2017).

Le TAL arabe est difficile parce que traditionnellement, les ordinateurs sont conçus pour que les humains leur « parlent » dans un langage de programmation précis, sans ambiguïté et extrêmement structuré, ou à l'aide d'un nombre limité de commandes vocales clairement énoncées. Or, le discours humain n'est pas toujours précis, il est souvent ambigu et sa structure linguistique peut dépendre d'un grand nombre de variables complexes, et limite le nombre des travaux existants de la classification des textes.

Notre objectif est de proposer un système de la classification des textes (arabes en particuliers) en utilisant différentes techniques (Naïve Bayésienne, Random Forest, Machine à Vecteurs de Support, Régression Logistique).

Le reste de ce mémoire est organisé en cinq chapitres :

Dans le deuxième chapitre intitulé « Background », nous présenterons les notions, outils et méthodes utilisés pour élaborer le projet et mettre en œuvre notre système.

Dans le troisième chapitre intitulé « Etat de l'art », nous présenterons les travaux existants dans le domaine de la classification des textes.

Dans le quatrième chapitre intitulé « Conception du système », nous présenterons les différentes étapes de conception de notre système.

Le cinquième chapitre, intitulé « Implémentation », introduit un guide d’utilisation avec un exemple illustratif pour faciliter la tâche aux utilisateurs de notre système.

Enfin, nous clôturons ce mémoire par une conclusion dans laquelle nous synthétisons notre travail, et nous exposons quelques perspectives futures.

Chapitre I : Background

1. Introduction

Dans ce chapitre nous introduisons les méthodes, outils et environnements qui nous ont aidés à mettre en œuvre notre système. On commence par TALN qui est le traitement automatique du langage Naturel. Ensuite, la classification des textes et bien sûr la notion de Machine Learning sont introduites. Enfin, la régression logistique et Machine à vecteurs de support sont détaillées.

2. Traitement Automatique du Langage Naturel (TALN)

2.1. Définition

Le Traitement Automatique du Langage Naturel (TALN) ou des langues (TAL) est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle, qui concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain. (Site 1, 2021)

Elle concerne la conception des systèmes et techniques informatiques permettant de manipuler le langage humain, dont le principal objectif est la conception et le développement de programmes capables de traiter de manière automatique des données linguistiques, c'est-à-dire des données exprimées dans un langage dit naturel (Ahmed et Rabah, 2019).

Ces dernières décennies, le traitement automatique des langues a connu une véritable ascension, que ce soit sur le plan scientifique mais aussi socio-économique, et cela par l'émergence de plusieurs firmes et de produits spécialisés. On parle aujourd'hui : de Traduction automatique, de correction automatique d'orthographe, de résumé automatique, d'interrogation de base de données en langues naturelle, etc.

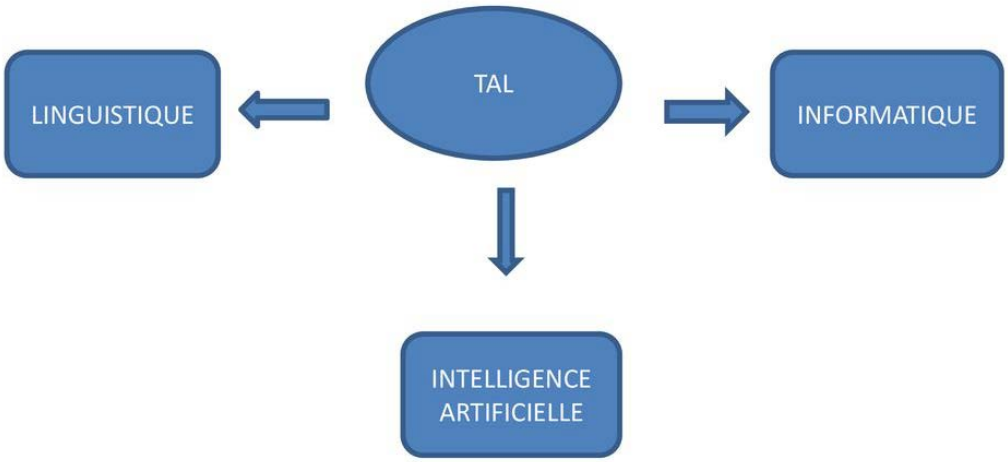

Figure 1: TAL est pluridisciplinaire

La réalisation de n'importe quelle application parmi celles citées précédemment passe principalement par différent niveaux (lexicale, morphologique, syntaxique, sémantique et pragmatique) mais aussi par le développement de plusieurs modules importants, où la réussite de l'application dépend pleinement de la performance de ces modules (Mohamed et Djilali, 2014).

2.2. Niveaux Traitement Automatique de la Langue

On va essayer de citer brièvement dans cette section les différents niveaux de traitement nécessaires pour parvenir à une compréhension complète d'un énoncé en langage naturel (Ahmed et Rabah, 2019).

Figure 2 schématise ces différents niveaux de traitements. Ces niveaux se superposent ; chacun apportant des problèmes spécifiques à résoudre relatifs à un niveau donné. En s'appuyant sur un découpage méthodologique classique dans le domaine de la linguistique, cela nous donne la hiérarchie suivante :

- **La phonétique** concerne l'étude des sons et prosodies (variations).
- **La phonologie** concerne l'étude de Phonèmes.
- **La morphologie** concerne l'étude de la formation des mots et de leurs variations de forme.
- **La syntaxe** consistant à extraire les relations grammaticales que les mots et groupes de mots entretiennent entre eux.
- **La sémantique** se consacre au sens des énoncés.
- **La pragmatique** prend en compte le contexte d'énonciation.

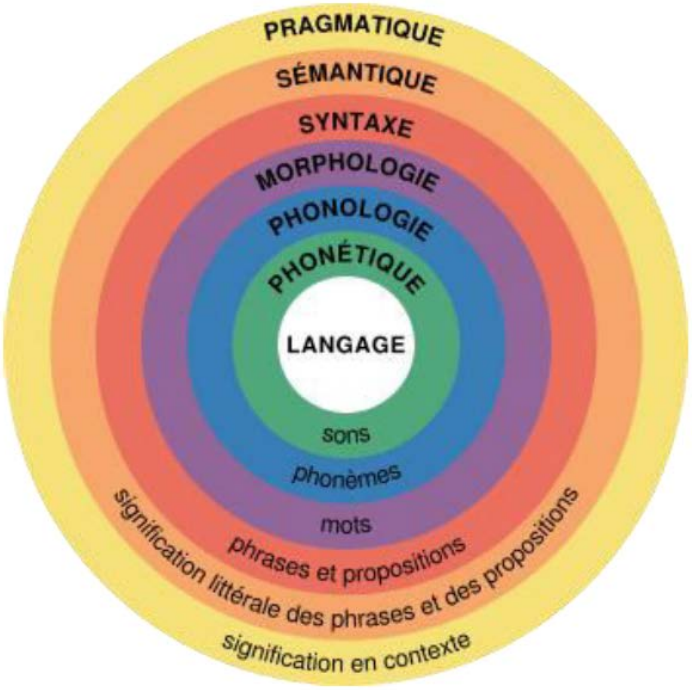

Figure 2 : Les niveaux de traitement (Ahmed et Rabah, 2019).

2.3. Traitement automatique de la langue arabe (TALA)

Le traitement automatique de la langue arabe est une discipline en pleine expansion, et dans laquelle on constate de plus en plus de recherches et de technologies qui portent un intérêt aux spécificités de cette langue et proposent des outils nécessaires au développement de son traitement automatique.

Le traitement automatique de l'arabe est un domaine de recherche stimulant. Il combine en effet plusieurs défis intéressants, parmi lesquels on peut citer la complexité morphologique de la langue, son haut degré d'ambiguïté et l'existence de nombreux dialectes présentant des variantes significatives.

Par ses propriétés morphologiques et syntaxiques, la langue arabe est considérée comme une langue difficile à maîtriser dans le domaine du traitement automatique de la langue.

Les recherches pour le traitement automatique de l'arabe ont débuté vers les années 1970. Les premiers travaux concernaient notamment les lexiques et la morphologie (Ahmed et Rabah, 2019).

Avec la diffusion de la langue arabe sur le Web et la disponibilité des moyens de manipulation de textes arabes, les travaux de recherche ont abordé des problématiques plus

variées comme la syntaxe, la traduction automatique, l'indexation automatique des documents, la recherche d'information, la catégorisation des textes, etc.

Le domaine du Traitement Automatique des Langues (TAL) appliqué à l'arabe a fait ces 15 dernières années des progrès considérables, mais il reste un grand chemin à faire pour pouvoir rivaliser avec d'autres langues comme le français et l'anglais (Ahmed et Rabah, 2019).

2.4. Quelques applications du TALN

Entre autre, on trouve les applications suivantes :

- Classification des textes en différentes catégories,
- Traduction automatique,
- Extraction d'information,
- Systèmes à question/réponse,
- L'interface homme-machine,
- La robotique,
- etc.

3. La classification des textes

Plusieurs définitions de la Classification des Textes ont vu le jour depuis son apparition. Nous citons dans ce contexte les deux définitions suivantes :

- **Définition 1 :** La classification de textes est un domaine où les algorithmes sont appliqués sur des documents de texte. Cette tâche consiste à attribuer un document dans une ou plusieurs classes, en fonction de son contenu. En règle générale, ces classes sont triées sur le volet par les humains. Certains applications populaires où la classification de textes est appliquée sont les suivantes (Chen et al., 2014) :
  - Classer les nouvelles comme Politique, Sports, Monde, Affaires, Style de vie, etc.
  - Classer les courriers électroniques comme Spam, Autre.
  - Classer Les documents de recherche par type de conférence.
  - Classer les critiques de films comme bons, mauvais et neutres.
  - Classer les blagues comme drôles, pas drôles.
- **Définition 2 :** Formellement, la classification de textes consiste à associer une valeur booléenne à chaque paire$(d_j, c_i) \in D \times C$, où "D" est l'ensemble des textes et "C" est

l'ensemble des catégories. La valeur V (Vrai) est alors associée au couple (d_j, c_i) si le texte d_j appartient à la classe c_i tandis que la valeur F (Faux) lui sera associée dans le cas contraire. Le but de la catégorisation de textes est de construire une procédure (modèle, classificateur) $\Phi : D \times C \rightarrow \{V, F\}$ qui associe une ou plusieurs catégories à un document d_j (Sebastiani, 2002).

3.1. But de la classification

Comme les autres méthodes de l'Analyse des données, dont elle fait partie, la Classification a pour but d'obtenir une représentation schématique simple d'un tableau rectangulaire de données dont les colonnes, suivant l'usage, sont des descripteurs de l'ensemble des observations, placées en lignes.

L'objectif le plus simple d'une classification est de répartir l'échantillon en groupes d'observations homogènes, chaque groupe étant bien différencié des autres. Le plus souvent, cependant, cet objectif est plus raffiné ; on veut, en général, obtenir des sections à l'intérieur des groupes principaux, puis des subdivisions plus petites de ces sections, et ainsi de suite. En bref, on désire avoir une hiérarchie, c'est à dire une suite de partitions "emboîtées", de plus en plus fines, sur l'ensemble d'observations initial (Maurice, 2006).

4. Machine Learning (ML)

4.1. Définition

Machine Learning est une branche de l'intelligence artificielle qui a pour but de donner la possibilité aux ordinateurs d'apprendre. Un ordinateur n'est pas intelligent, il ne fait qu'exécuter des tâches. On lui décrit sous forme de programmes quoi faire et comment le faire. C'est ce qu'on appelle la programmation (Younes, 2016).

Machine Learning traite des sujets complexes où la programmation traditionnelle trouve ses limites. Construire un programme qui conduit une voiture serait très complexe voire impossible. Cela étant dû aux nombres infinis des cas possibles à traiter… ML traite cette problématique différemment. Au lieu de décrire quoi faire, le programme apprendra par lui même comment conduire en "observant" des expérimentations.

Machine Learning : Donner la possibilité à l'ordinateur d'apprendre sans être programmé.

En fonction des données d'expérimentation que prendra l'algorithme d'apprentissage en entrée, il déduira par lui même une hypothèse de fonctionnement. Il utilisera cette dernière pour de nouveaux cas, et affinera son expérience au fil du temps.

4.2. Types de problèmes en Machine Learning

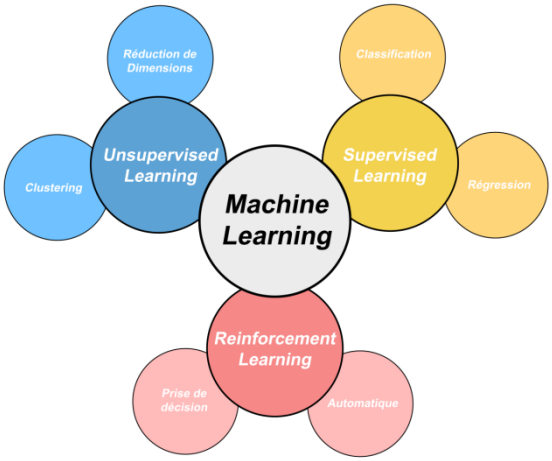

Figure 3 : Types de problèmes en Machine Learning (Guillaume, 2019)

- ✓ **Apprentissage Supervisé :** C'est le type le plus récurrent. Il s'agit de fournir aux algorithmes d'apprentissages un jeu de données d'apprentissage (Training Set) sous forme de **(X, Y)** avec X les variables prédictives, et Y le résultat de l'observation. En se basant sur le *Training Set*, l'algorithme va trouver **une fonction mathématique** qui permet de transformer (au mieux) X vers Y. En d'autres termes, l'algorithme va trouver une fonction F tel que : $F(X) \approx Y$ (Younes, 2016).

  Il existe plusieurs algorithmes et techniques utilisés pour la classification supervisée telles que :

  - Classificatiou Bayésienne
  - Machine à vecteurs de support (SVM)
  - Réseau neuronaux
  - Forêts d'arbres décisionnels (Random Forest)
  - Le Boosting
- ✓ **Apprentissage Non supervisé :** Dans ce type, on va donner à l'algorithme des données, éventuellement **non structurées**. Et on le laisse trouver une sorte de **structure** dans nos données. Cela peut être des regroupements de données (Clustering).

Il existe plusieurs algorithmes et techniques utilisés pour la classification non supervisée, telles que :

- K-moyennes (KMeans)
- Fuzzy
- Espérance-Maximisation (EM)
- Regroupement hiérarchique

✓ **Apprentissage par renforcement :** Pour guider l'apprentissage automatique non supervisé, certains acteurs utilisent une troisième méthode : l'apprentissage par renforcement. Elle consiste à introduire un système de récompenses et de punitions afin d'induire le comportement et les décisions (ou actions) d'un algorithme (ou agent) dans un environnement donné, réel ou virtuel. Cette technique, moins répandue parce que bien plus complexe, est utilisé dans les domaines de la recherche opérationnelle, de la théorie des jeux, de la théorie du contrôle, de l'optimisation fondée sur la simulation ou encore dans les statistiques génétiques. Ici, il n'y a pas de jeux de données en entrée. L'algorithme construit son dataset en exploitant l'environnement. (Site 2, 2021)

4.3. Importance de Machine Learning interprétable par des humains

Lorsqu'un modèle de ML est complexe, il peut s'avérer difficile d'en expliquer le fonctionnement. Dans certains secteurs spécialisés, les data-scientists doivent utiliser des modèles de Machine Learning simples, car il est important pour l'entreprise d'expliquer en détail comment chaque décision a été prise. C'est notamment le cas dans les secteurs soumis à de fortes exigences de conformité, tels que la banque et l'assurance. Dans ces secteurs hautement régulés, les équipes de data-science ont souvent l'obligation de fournir une documentation détaillée sur les modèles déployés.

Les modèles complexes peuvent donner des prédictions précises, mais il peut être difficile d'expliquer à une personne non initiée comment un résultat a été obtenu.

5. La régression logistique (LR)

5.1. Définition

La régression logistique (Logistic regression) est un algorithme supervisé de **classification**, populaire en Machine Learning. (Younes, 2018).

5.2. La classification en Machine Learning

La classification est une tâche très répandue en Machine Learning. Il existe deux types :

- Binary Classification (classification binaire): l'étiquette Y aura deux valeurs possibles 0 ou 1. En d'autres termes $Y \in \{0,1\}$.

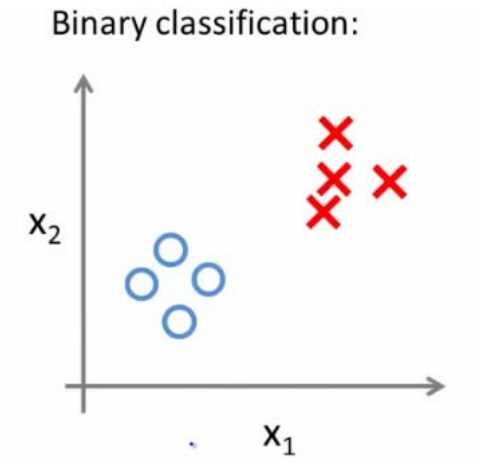

Figure 4 : Classification binaire

- Multi-class classification (Classification Multi-classe): $Y \in \{0,1,2, \dots\}$

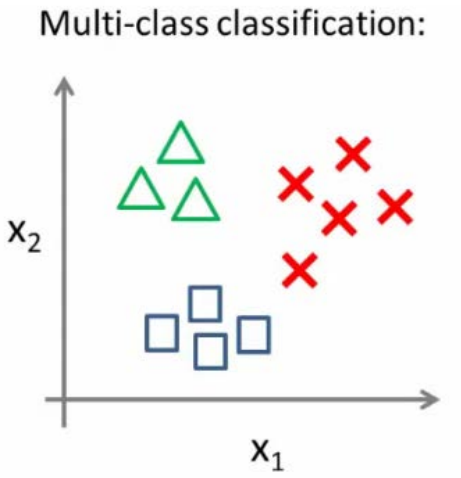

Figure 5 : Classification Multi-classe

5.3. La fonction sigmoid (Sigmoid Function)

La fonction *score* qu'on a obtenue intègre les différentes variables prédictives (lesx_i). A cette fonction, on appliquera **la fonction sigmoid** (**Sigmoid Function)**. Cette fonction produit des valeurs comprises entre 0 et 1.

Le résultat obtenu par la fonction sigmoid est interprété comme la **probabilité que l'observation X soit d'un label (étiquette) 1.**

La fonction Logistique (autre nom pour la fonction Sigmoid), est définie comme suit

$$Sigmoid(x) = \frac{1}{1 + e^{-x}}$$

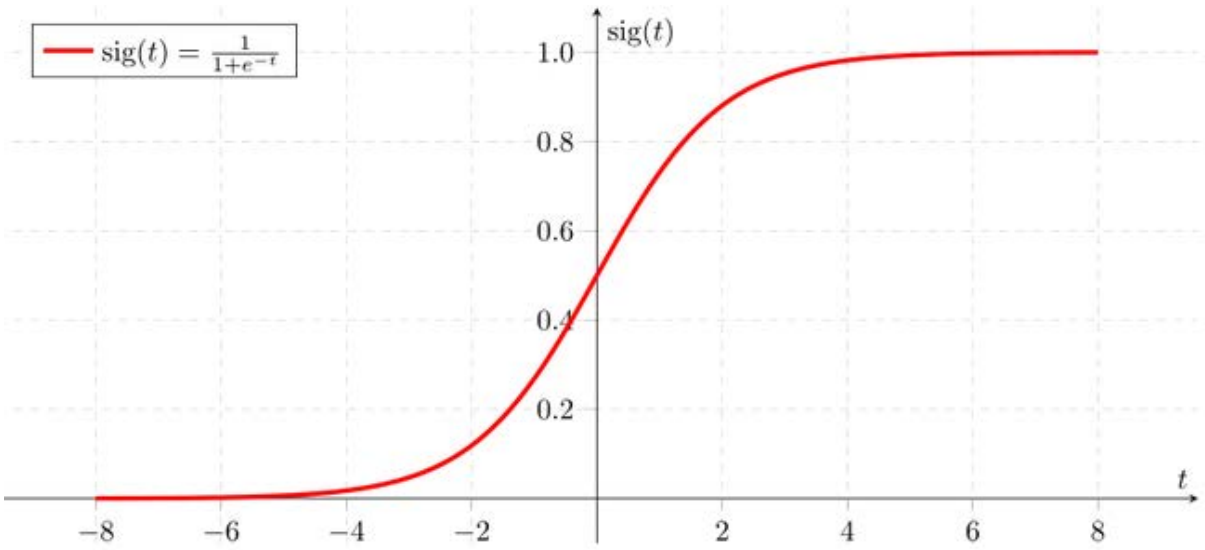

Figure 6 : La fonction sigmoid

6. Machine à vecteurs de support (SVM)

Les machines à support de vecteurs (SVM) sont à l'origine de nouvelles méthodes de catégorisations, bien que les premières publications sur le sujet datent des années 60 (Hanane, 2018).

Avant d'aborder le principe de fonctionnement général des SVM voici quelques notions de base :

- **Hyperplan** : est un séparateur d'objets des classes. De cette notion, nous pouvons dire qu'il est évident de trouver une mainte d'hyperplans mais la propriété délicate des SVM est d'avoir l'hyperplan dont la distance minimale aux exemples d'apprentissage est maximale, cet hyperplan est appelé L'hyperplan optimal, et la distance appelée marge.
- **Vecteurs Support** : ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches de ce dernier.

Voici un schéma représentatif de ces notions (Raheel, 2010):

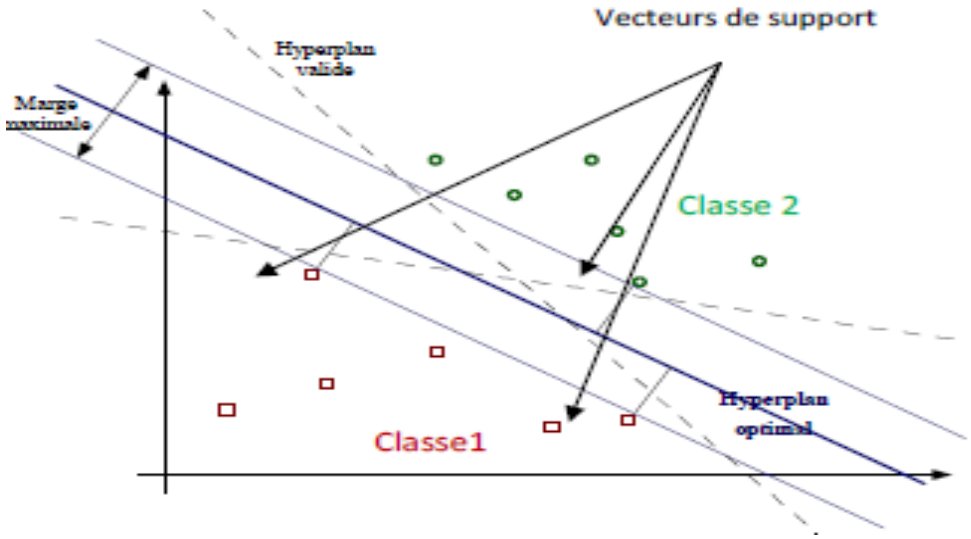

Figure 7 : Vecteurs à support

Le principe des SVM consiste en une stratégie de minimisation structurelle du risque mais le problème revient à trouver une frontière de décision qui sépare l'espace en deux régions, à trouver l'hyperplan qui classifie correctement les données et qui se trouve le plus loin possible de tous les exemples. On dit qu'on veut maximiser la marge qui veut dire la distance du point le plus proche de l'hyperplan (SAHRAOUI, 2013).

Dans le cas de la catégorisation des textes, les entrées sont des documents et les sorties sont des catégories. En considérant un classificateur binaire, on voudra lui faire apprendre l'hyperplan qui sépare les documents appartenant à la catégorie et ceux qui n'en font pas partie (DERDRA et BENSFIA, 2012).

Les SVM conviennent bien pour la classification de textes parce qu'une dimension élevée ne les affecte pas puisqu'ils se protègent contre le sur apprentissage. Autrement dit, il affirme que peu d'attributs sont totalement inutiles à la tâche de classification et que les SVM permettent d'éviter une sélection agressive qui aurait comme résultat une perte d'information. On peut se permettre de conserver plus d'attributs. Également, une caractéristique des documents textuels est que lorsqu'ils sont représentés par des vecteurs, une majorité des entrées sont nulles.

Or, les SVM conviennent bien à des vecteurs dits clairsemés. Un autre aspect positif des SVM est qu'aucun ajustement de paramètres manuel n'est requis, car ils ont l'habileté de trouver automatiquement des paramètres adéquats (Simon, 2005).

6.1. Avantages et inconvénients des machines à vecteurs de support

- Avantages
  - ✓ Sa grande précision de prédiction.

- ✓ Fonctionne bien sur de plus petits data sets.
- ✓ Ils peuvent être plus efficaces car ils utilisent un sous-ensemble de points d'entraînement.

- Inconvénient
  - ✓ Ne convient pas à des jeux de données plus volumineux, car le temps d'entraînement avec les SVM peut être long.
  - ✓ Moins efficace sur les jeux de données contenant du bruits et beaucoup d'outliers.

7. Forêts d'arbres décisionnels (Random Forest)

7.1. Définition

Forêt d'arbres décisionnels est un algorithme de classification qui réduit la variance des prévisions d'un arbre de décision seul, améliorant ainsi leurs performances. Pour cela, il combine de nombreux arbres de décisions dans une approche de type bagging. (Site 3, 2021)

Les forêts d'arbres décisionnels (Robert et al., 2009) ont été premièrement proposées par Ho en 1995(Ho et Tin,1995) et ont été formellement proposées en 2001 par Leo Breiman (Leo , 2001) et Adele Cutler (Andy, 2012) . Elles font partie des techniques d'apprentissage automatique. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

7.2. Avantages et inconvénients des Forêts d'arbres décisionnels

- Avantages
  - ✓ Reconnaissance TRES RAPIDE
  - ✓ Multi-classes par nature
  - ✓ Efficace sur inputs de grande dimension
  - ✓ Robustesse aux outliers
- Inconvénients
  - ✓ Apprentissage souvent long
  - ✓ Valeurs extrêmes souvent mal estimées dans cas de régression

8. Classification naïve bayésienne

La classification naïve bayésienne est un type de classification Bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classifieur bayésienne naïf, ou classifieur naïf de Bayes, appartenant à la famille des classifieurs Linéaires (Choayb, 2014).

Un terme plus approprié pour le modèle probabiliste sous-jacent pourrait être « modèle à Caractéristiques statistiquement indépendantes » (Harry, 2004).

En termes simples, un classifieur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques. Un fruit peut être considéré comme une pomme s'il est rouge, arrondi, et fait une dizaine de centimètres. Même si ces caractéristiques sont liées dans la réalité, un classifieur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille.

Selon la nature de chaque modèle probabiliste, les classifieurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé.

Dans beaucoup d'applications pratiques, l'estimation des paramètres pour les modèles bayésiennes naïfs repose sur le maximum de vraisemblance. Autrement dit, il est possible de travailler avec le modèle bayésienne naïf sans se préoccuper de probabilité bayésienne ou utiliser les méthodes bayésiennes.

Malgré leur modèle de conception « naïf » et ses hypothèses de base extrêmement simplistes, les classifieurs bayésienne naïfs ont fait preuve d'une efficacité plus que suffisante dans beaucoup de situations réelles complexes. En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue, (Harry, 2004). Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats (Caruana et Niculescu, 2006).

L'avantage du classifieur bayésienne naïf est qu'il requiert relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification, à savoir moyennes et variances des différentes variables. En effet, l'hypothèse d'indépendance des variables permet de se contenter de la variance de chacune d'entre elle pour chaque classe, sans avoir à calculer de matrice de covariance.

8.1. Description du modèle Bayésienne

Le modèle probabiliste pour un classifieur est le modèle conditionne $p(C|F1, \dots, F\mathfrak{y})$

où C est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques F1,…, F$\mathfrak{y}$ (Choayb, 2014).

Lorsque le nombre de caractéristiques $\mathfrak{y}$ est grand, ou lorsque ces caractéristiques peuvent prendre un grand nombre de valeurs, baser ce modèle sur des tableaux de probabilités devient impossible (Caruana et Niculescu, 2006).

Par conséquent, nous le dérivons pour qu'il soit plus facilement soluble. À l'aide du théorème de Bayes, nous écrivons (Choayb, 2014).

$$p(C|F_1, \dots, F_n) = \frac{p(C)\ p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)}.$$

En langage courant, cela signifie :

$$\text{postérieure} = \frac{\text{antérieure} \times \text{vraisemblance}}{\text{évidence}}.$$

En pratique, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de C et les valeurs des caractéristiques Fi sont données. Le dénominateur est donc en réalité constant. Le numérateur est soumis à la loi de probabilité à plusieurs variables. P(C,F1,…,F$\mathfrak{y}$)

et peut être factorisé de la façon suivante, en utilisant plusieurs fois la définition de la probabilité conditionnelle :

$$\begin{aligned}
&p(C, F_1, \dots, F_n) \\
&\quad = p(C)\ p(F_1, \dots, F_n|C) \\
&\quad = p(C)\ p(F_1|C)\ p(F_2, \dots, F_n|C, F_1) \\
&\quad = p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3, \dots, F_n|C, F_1, F_2) \\
&\quad = p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ p(F_4, \dots, F_n|C, F_1, F_2, I
\end{aligned}$$

C'est là que nous faisons intervenir l'hypothèse naïve : si chaque Fi est indépendant des autres caractéristiques yi ≠ j alors

Pour tout $i \neq j$, par conséquent la probabilité conditionnelle peut s'écrire

$$p(F_i|C, F_j) = p(F_i|C)$$

$$p(C, F_1, \ldots, F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots$$

$$= p(C) \prod_{i=1}^{n} p(F_i|C).$$

Par conséquent, en tenant compte de l'hypothèse indépendance ci-dessus, la probabilité conditionnelle de la variable de classe C peut être exprimée par où

Où Z (appelé « évidence ») est un facteur d'échelle qui dépend uniquement de F1,…, F**ŋ**, à savoir une constante dans la mesure où les valeurs des variables caractéristiques sont connues.

Les modèles probabilistes ainsi décrits sont plus faciles à manipuler, puisqu'ils peuvent être factorisés par l'antérieure P(C) (probabilité a priori de C) et les lois de probabilité indépendantes P(Fi|C). S'il existe K classes pour C et si le modèle pour chaque fonction peut être exprimé selon paramètres, alors le modèle bayésien naïf correspondant dépend de (k − 1) + n r k paramètres.

Dans la pratique, on observe souvent des modèles où K=2 (classification binaire) et r=1 (les caractéristiques sont alors des variables de Bernoulli). Dans ce cas, le nombre total de paramètres du modèle bayésien naïf ainsi décrit est de 2n+1, avec n le nombre de caractéristiques binaires utilisées pour la classification.

8.2. Estimation de la valeur des paramètres

Tous les paramètres du modèle (probabilités a priori des classes et lois de probabilités associées aux différentes caractéristiques) peuvent faire l'objet d'une approximation par rapport aux fréquences relatives des classes et caractéristiques dans l'ensemble des données d'entraînement. Il s'agit d'une estimation du maximum de vraisemblance des probabilités. Les probabilités a priori des classes peuvent par exemple être calculées en se basant sur l'hypothèse que les classes sont équiprobables (i.e chaque antérieure = 1 / (nombre de classes)), ou bien en estimant chaque probabilité de classe sur la base de l'ensemble des données d'entraînement (i.e antérieure de C= (nombre d'échantillons de C) / (nombre d'échantillons total)).

Pour estimer les paramètres d'une loi de probabilité relative à une caractéristique précise, il est nécessaire de présupposer le type de la loi en question ; sinon, il faut générer des modèles non-paramétriques pour les caractéristiques appartenant à l'ensemble de données d'entraînement. Lorsque l'on travaille avec des caractéristiques qui sont des variables aléatoires continues, on suppose généralement que les lois de probabilités correspondantes sont des lois normales, dont on estimera l'espérance et la variance.

L'espérance, μ, se calcule avec $$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Où N est le nombre d'échantillons et xi est la valeur d'un échantillon donné.

La variance, σ^2, se calcule avec

$$\sigma^2 = \frac{1}{(N-1)}\sum_{i=1}^{N} (x_i - \mu)^2$$

Si, pour une certaine classe, une certaine caractéristique ne prend jamais une valeur donnée dans l'ensemble de données d'entraînement, alors l'estimation de probabilité basée sur la fréquence aura pour valeur zéro. Cela pose un problème puisque l'on aboutit à l'apparition d'un facteur nul lorsque les probabilités sont multipliées. Par conséquent, on corrige les estimations de probabilités avec des probabilités fixées à l'avance.

9. Conclusion

Dans ce chapitre, nous avons vu les notions, outils et méthodes utilisés pour élaborer notre projet. Passant maintenant au chapitre suivant concernant l’état de l’art.

Chapitre II : Etat de l'art

1. Introduction

Dans ce chapitre, nous présenterons les travaux et les outils existants dans le domaine de la classification des textes.

2. Quelques travaux et outils existants pour l'Anglais

La langue anglaise est la plus parlée au monde ; elle se classe troisième, après le chinois et l'espagnol. Plusieurs travaux de recherches sur la classification des textes anglais on été proposés.

1- **Travaux de Xiaoyu Luo (2021)** : il a mis en œuvre le modèle Support Vector Machines (SVM) pour classer le texte et les documents anglais. Ici, il a fait deux expériences analytiques pour vérifier les classificateurs sélectionnés à l'aide de documents anglais. Les résultats expérimentaux effectués sur un ensemble de 1033 documents texte montrent que le classificateur Rocchio fournit les meilleurs résultats de performances lorsque la taille de l'ensemble de fonctionnalités est petite tandis que SVM surpasse les autres classificateurs. A partir de l'analyse expérimentale, nous avons observé que le taux de classification dépasse 90 % lorsque l'on utilise plus de 4000 fonctionnalités (Xiaoyu, 2021).

2- **Travaux de Yujia Bao et al. (2020)** : ils explorent le méta-apprentissage pour la classification des textes en quelques plans. leur modèle est entraîné dans un cadre de méta-apprentissage pour mapper ces signatures en scores d'attention, qui sont ensuite utilisés pour pondérer les représentations lexicales des mots (Yujia et al., 2020).

3- **Travaux de Santiago et Eduardo (2020)** : ils présentent BERT et une revue des approches classiques de la PNL et testent empiriquement avec une suite de différents scénarios le comportement de BERT par rapport au vocabulaire TF-IDF traditionnel fourni aux algorithmes d'apprentissage automatique (Santiago et Eduardo, 2020).

4- **Travaux de Sebastian et al. (2021)** : ce travail compare trois représentations de texte créées manuellement et des intégrations de texte créées automatiquement par les outils AutoML. Leur référence comprend quatre outils AutoML open source populaires et huit ensembles de données à des fins de classification de texte (Sebastian et al., 2021).

5- **Travaux de Shervin et al. (2020)** : ils fournissent un examen complet de plus de 150 modèles basés sur l'apprentissage en profondeur pour la classification de texte développés ces dernières années, et discutent de leurs contributions techniques, similitudes et forces et

fournissent également un résumé de plus de 40 ensembles de données populaires largement utilisés pour la classification des textes (Shervin et al., 2020).

6- **Travaux de Ankit et al. (2020)** : dans ce travail, un modèle basé sur un réseau d'attention graphique est proposé pour capturer la structure de dépendance attentive parmi les étiquettes. Le réseau d'attention graphique utilise une matrice de caractéristiques et une matrice de corrélation pour capturer et explorer les dépendances cruciales entre les étiquettes et générer des classificateurs pour la tâche. Les classificateurs générés sont appliqués aux vecteurs de caractéristiques de phrases obtenus à partir du réseau d'extraction de caractéristiques de texte (BiLSTM) pour permettre un apprentissage de bout en bout. L'attention permet au système d'attribuer différents poids aux nœuds voisins par étiquette, lui permettant ainsi d'apprendre implicitement les dépendances entre les étiquettes. Les résultats du modèle proposé sont validés sur cinq ensembles de données MLTC du monde réel. Le modèle proposé atteint des performances similaires ou meilleures par rapport aux modèles de pointe précédents (Ankit et al., 2020).

Le tableau suivant montre quelques travaux et outils existants pour la langue anglaise:

Auteurs	**Machine Learning**	**Deep Learning**	**Outils**	**Métrique d'évaluation**
Xiaoyu, 2021	X		SVM	Accuracy – F1
Yujia et al., 2020		X	CNN	Accuracy
Santiago et Eduardo, 2020	X		BERT	Accuracy
Sebastian et al., 2021	X		AutoML	Accuracy
Shervin et al., 2020		X	-	-
Ankit et al., 2020		X	GNN	Micro-F1

Tableau 1 : Travaux et outils existants pour l'Anglais

3. Quelques travaux et outils existants pour l'Arabe

1- Travaux de WISSAME et NASSIMA (2020) : dans ce travail, ils ont conçu et réalisé un système qui permet de traiter des commentaires édités en arabe. Ils ont utilisé trois algorithmes pour analyser et classer les commentaires personnels sur un sujet particulier dans différents domaines. Les catégories qu'ils ont définies sont : positives et négatives (WISSAME et NASSIMA, 2020).

2- Travaux de Shammur et al. (2020) : ils utilisent une catégorisation des textes étiquetant les publications sur les réseaux sociaux dans des catégories telles que «sports», «politique», «droits de l'homme», entre autres, pour montrer l'efficacité des modèles à travers différentes sources et variétés d'arabe (Shammur et al., 2020).

3- Travaux de Ahlam et al. (2021) : dans ce travail, trois méthodes différentes ont été utilisées. La première méthode, Survey / Systematic Review, pour connaître les enjeux. La deuxième méthode, revue de la littérature, pour examiner l'effet de l'utilisation d'algorithmes d'apprentissage en profondeur pour classer le texte arabe. La troisième méthode consiste à expérimenter pour trouver un algorithme efficace qui rétablit une meilleure précision (Ahlam et al., 2021).

4- Travaux de Imad et Anoual (2021) : dans ce travail, on étudie l'impact de techniques de prétraitement judicieusement sélectionnées sur l'efficacité de différents algorithmes de classification des textes. Les effets de la suppression des mots vides, du radicalisme, de la lemmatisation et de toutes les combinaisons possibles sont examinés (Imad et Anoual, 2021).

5- Travaux de Samir et al. (2018) : dans ce travail, ils présentent une méthode innovante pour la classification des textes arabes. ils utilisent un algorithme de radicalisation arabe pour extraire, sélectionner et réduire les caractéristiques dont ils ont besoin. Après cela, ils utilisent la technique Term Frequency-Inverse Document Frequency comme technique de pondération des caractéristiques. Et enfin, pour l'étape de classification, ils utilisent l'un des algorithmes d'apprentissage profond qui est très puissant dans d'autres domaines comme le traitement d'images et la reconnaissance de formes, mais encore rarement utilisé en text mining ; cet algorithme est le Convolutional Neural Networks. Avec cette combinaison et quelques réglages d'hyperparamètres dans l'algorithme Convolutional Neural Networks, ils ont pu obtenir d'excellents résultats sur plusieurs benchmarks (Samir et al., 2018).

6- Travaux de Rasha et Mahmoud (2017) : ce travail présente la nature complexe de la langue arabe, pose les problèmes de manque de corpus arabes publics gratuits, explique les phases de classification qui jettent la littérature sur la classification des textes arabes (Rasha et Mahmoud, 2017).

7- Travaux de Adel Hamdan (2019) : ce travail utilise les algorithmes les plus connus utilisés dans la classification des textes avec des ensembles de données arabes. En plus de cela, l'ensemble de données utilisé est suffisamment volumineux par rapport à la plupart des ensembles de données pour la langue arabe utilisés dans d'autres recherches. En outre, utiliser différentes sélections et méthodes de pondération pour les documents (Adel Hamdan, 2019).

Le tableau suivant montre quelques travaux et outils existants pour la langue Arabe :

Auteurs	Machine Learning	Deep Learning	Outils	Métrique d'evaluation
WISSAME et NASSIMA, 2020	X		KNN, NB, SVM	Accuracy
Shammur et al., 2020	X			Macro F1
Ahlam et al., 2021		X	----------	----------
Imad et Anoual, 2021	X		NB, SVM, DT	
Samir et al., 2018		X	CNN	Accuracy
Rasha et Mahmoud, 2017	X		----------	----------
Adel Hamdan, 2019	X		NB, SVM, KNN, DT, Racchio	Precision, Recall

Tableau 2 : Travaux et outils existants pour l'Arabe

4. Conclusion

Dans ce chapitre, nous avons vu les travaux et les outils existants dans le domaine de la classification des textes. Passant maintenant au chapitre suivant concernant la conception du système.

Chapitre III : Conception du système

1. Introduction

Nous présentons dans ce chapitre la partie conception de notre système de classification supervisée des textes arabes. La conception était faite en langage de modélisation UML. Parmi les diagrammes UML, nous avons choisi ceux que nous avons jugés nécessaires pour la mise en œuvre de notre application. Pour modéliser l'aspect fonctionnel, nous avons opté pour le diagramme de cas d'utilisation. Pour modéliser l'aspect dynamique, nous avons choisi le diagramme d'activités et le diagramme de séquence.

2. Diagramme de cas d'utilisation

La figure 8 présente le diagramme de cas d'utilisation définissant les exigences fonctionnelles attendues, les acteurs (utilisateurs du système) ainsi que les relations qui unissent les acteurs et les fonctionnalités :

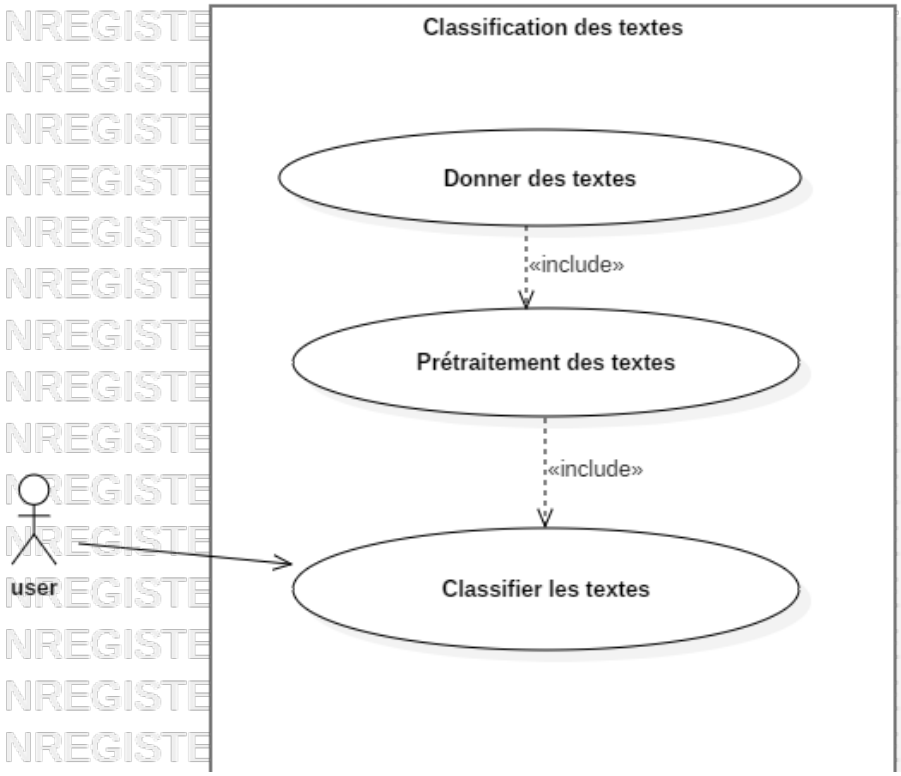

Figure 8 : Diagramme de cas d'utilisation

3. Diagramme d’activités

La figure 9 est le diagramme d’activité entre l’utilisateur et notre système :

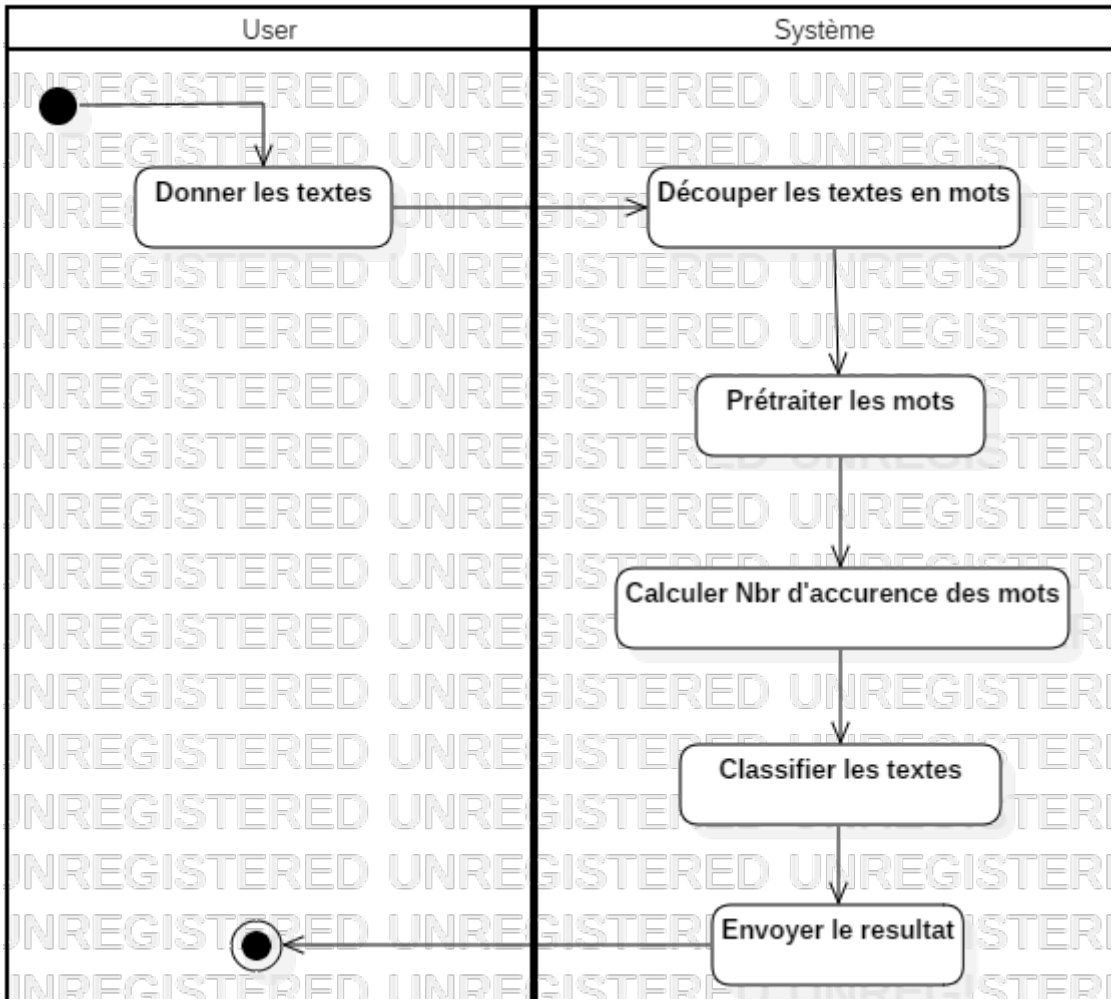

Figure 9 : Diagramme d’activités

4. Diagramme de séquence

La figure 10 présente la séquence des opérations permettant de réaliser le cas d'utilisation «classification des textes» :

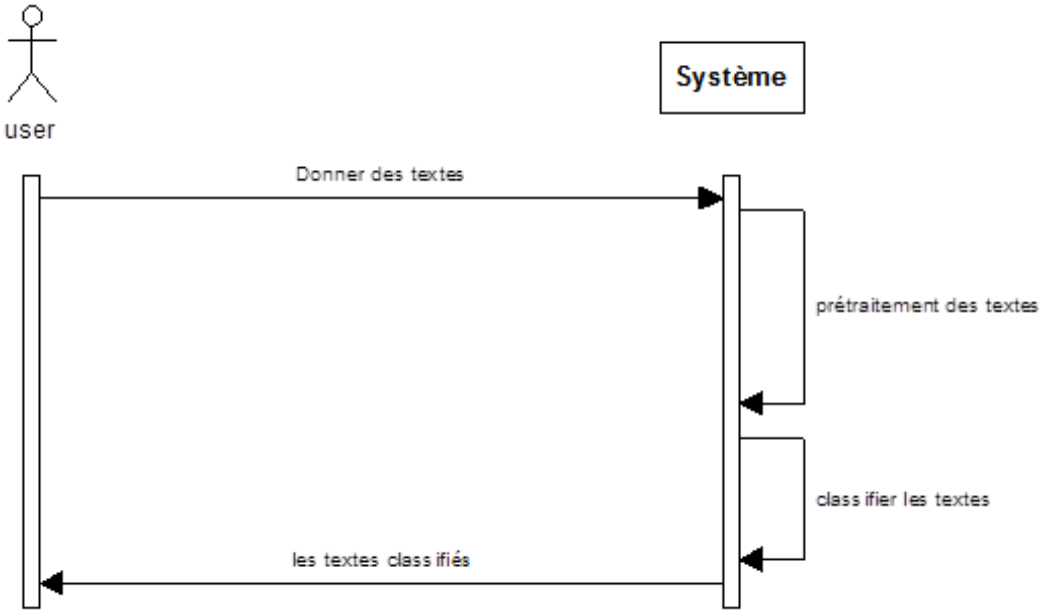

Figure 10 : Diagramme de séquence

5. Conclusion

Dans ce chapitre, nous avons détaillé la conception de notre application. Le chapitre suivant mettra en évidence, le fruit de ce passage et les différents résultats du développement de l'application demandée.

Chapitre IV : Implémentation

1. Introduction :

Ce chapitre présente l'architecture et la méthode d'utilisation de notre système, tout en présentant un exemple d'application. Ceci donne un guide aux utilisateurs pour bien utiliser le logiciel. En plus, on décrit les stratégies qu'on a utilisées pour réaliser notre travail.

2. Architecture de notre système

La figure suivante montre l'architecture de notre système :

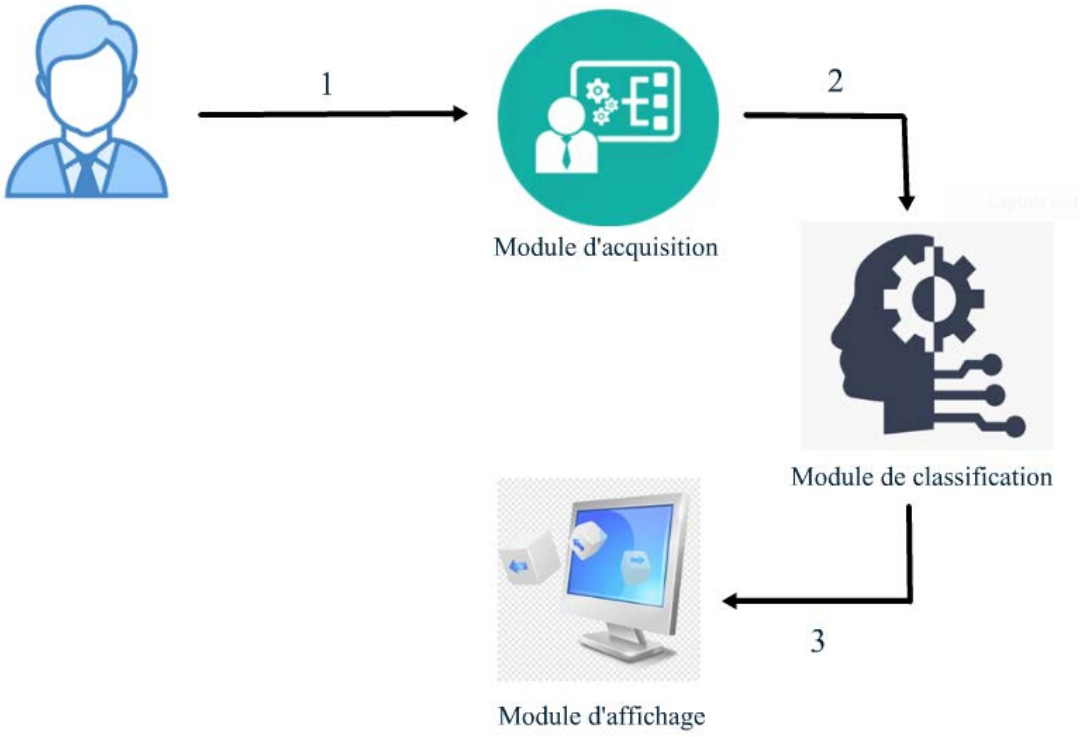

Figure 11 : Architecture de notre système

Le système contient trois grands modules : module d'acquisition, module de classification et module d'affichage. Dans ce qui suit, on décrit ces trois composantes :

1. Module d'acquisition : l'utilisateur introduit le texte.

2. Module de classification: c'est classifier le texte.

3. Module d'affichage : permet d'afficher le résultat de la classification.

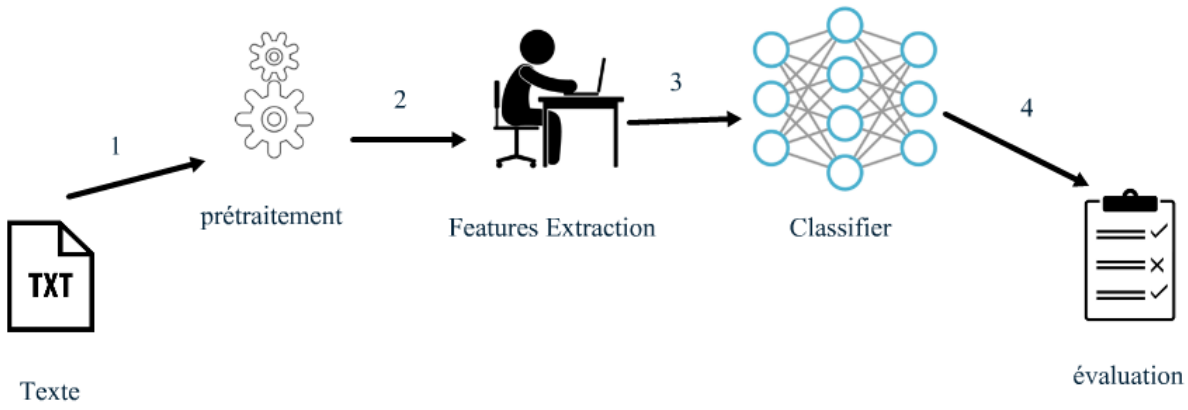

Figure 12 : Module de classification

Le module de classification contient: prétraitement, Features Extraction, Classifier et évaluation:

1- Prétraitement : consiste généralement en une segmentation de mots, un nettoyage des données et des statistiques de données.
2- Features Extraction : la représentation de texte vise à exprimer du texte prétraité sous une forme beaucoup plus simple pour les ordinateurs et minimisant la perte d'informations, comme le sac de mots (BOW), le N-gramme, le terme fréquence-fréquence inverse du document (TF-IDF), etc.
3- Classifier : le texte représenté est introduit dans le classificateur en fonction des caractéristiques sélectionnées (comme SVM, NB, etc.)
4- Evaluation : accuracy et le score F1 sont les plus utilisés pour évaluer les méthodes de classification de texte.

3. Présentation de l'application

La figure 13 représente l'interface principale de notre application. Dans cette fenêtre existe le menu Fichier.

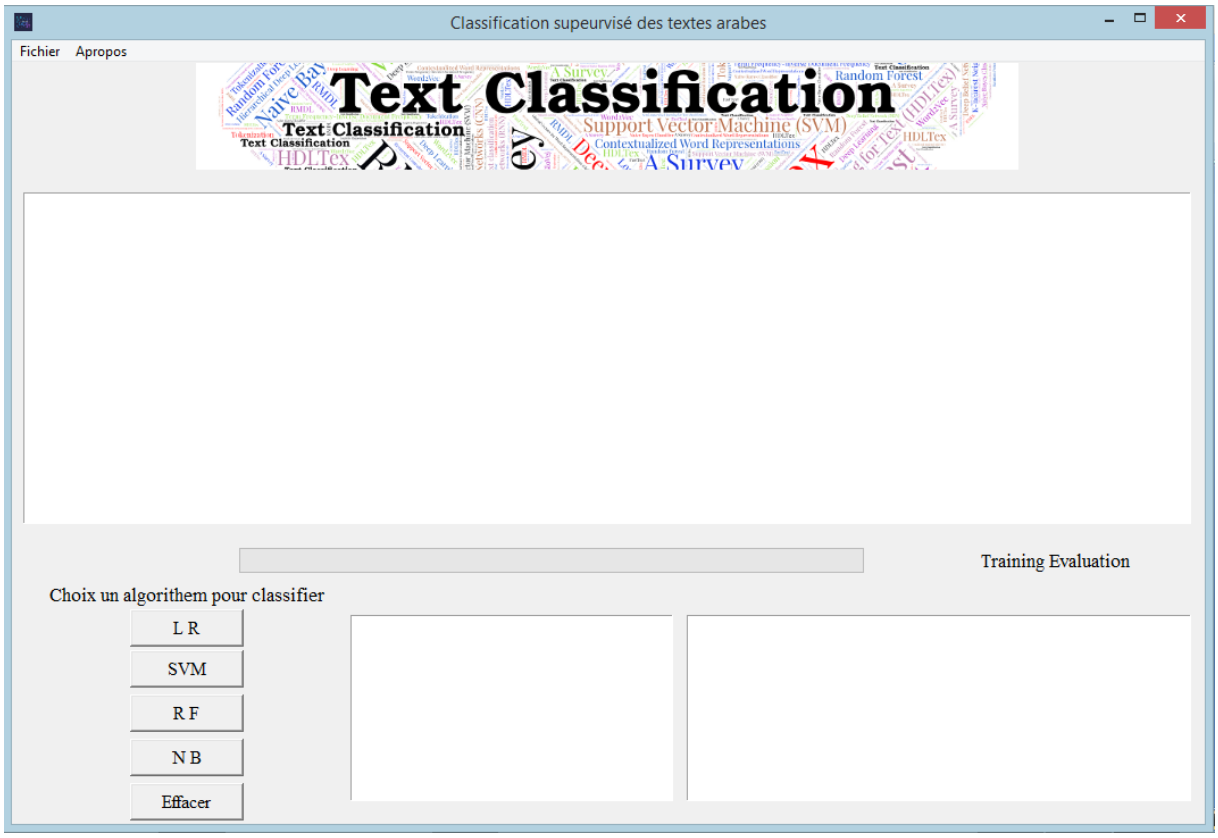

Figure 13 : l'interface principale

Dans le menu « Fichier», on trouve «Nouveau» qui permet de créer un nouveau projet.

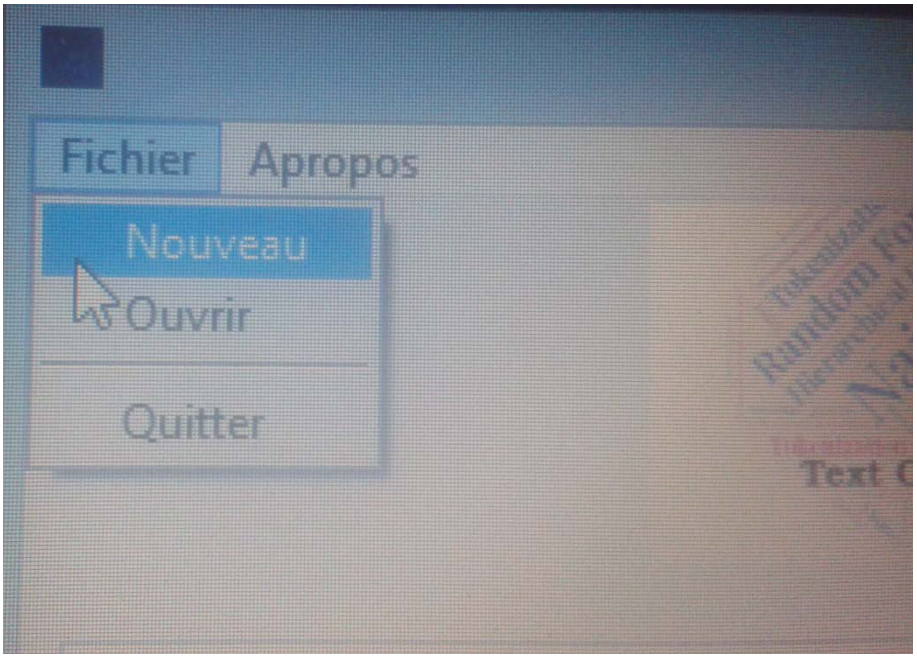

Figure 14 : Le menu Ficher

Lorsqu'on clique sur «ouvrir», la fenêtre suivante s'affiche (Figure 15).

Choix le nom du ficher texte et cliquer sur « Ouvrir ».

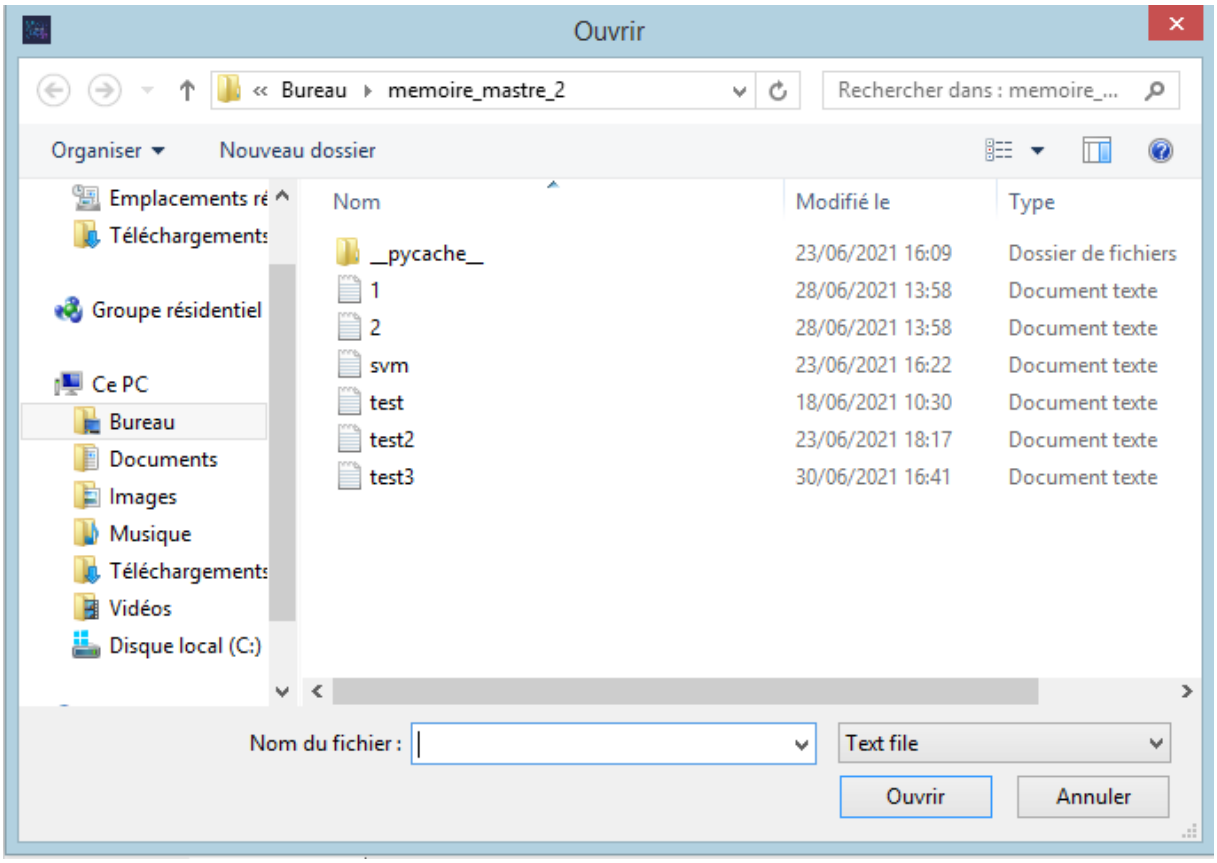

Figure 15 : fenêtre d'ouverture d'un nouveau fichier texte

Pour fermer logiciel cliquer sur « Quitter ».

Dans le menu « A propos», on trouve une fenêtre qui contient quelques informations sur le logiciel.

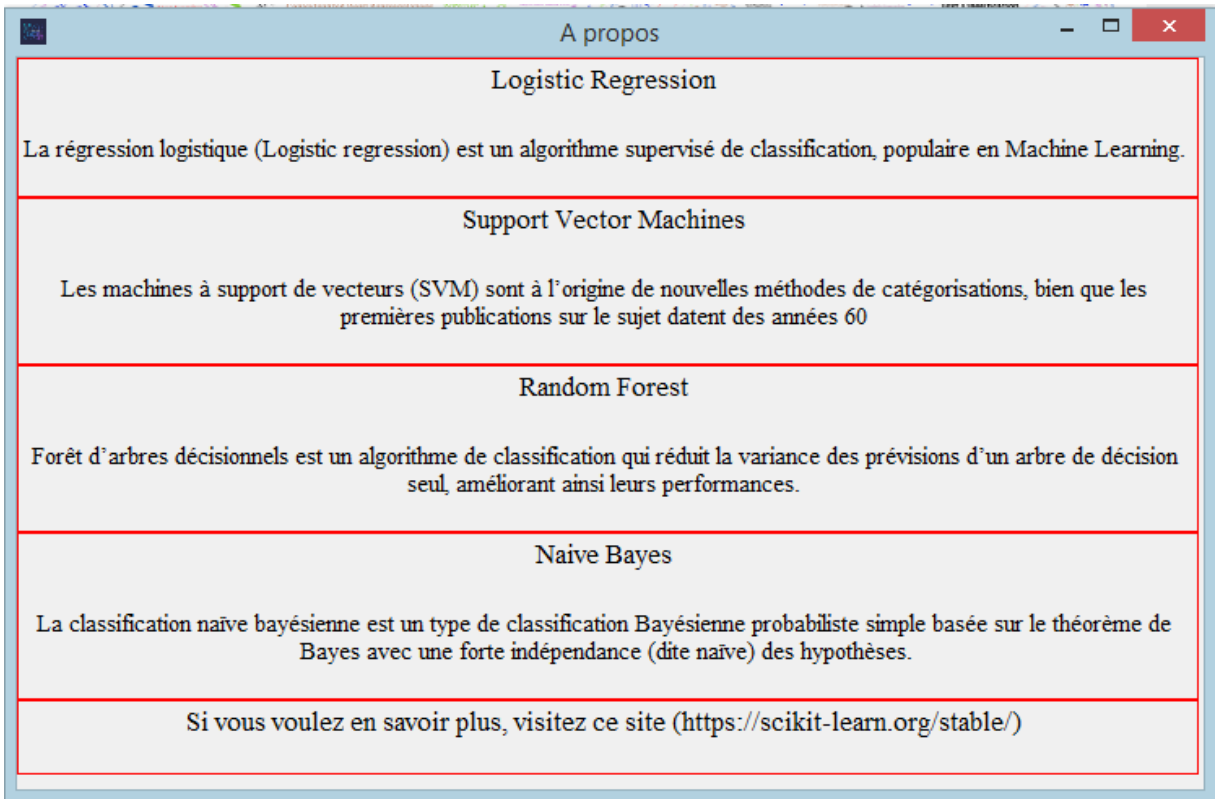

Figure 16 : fenêtre A-propos

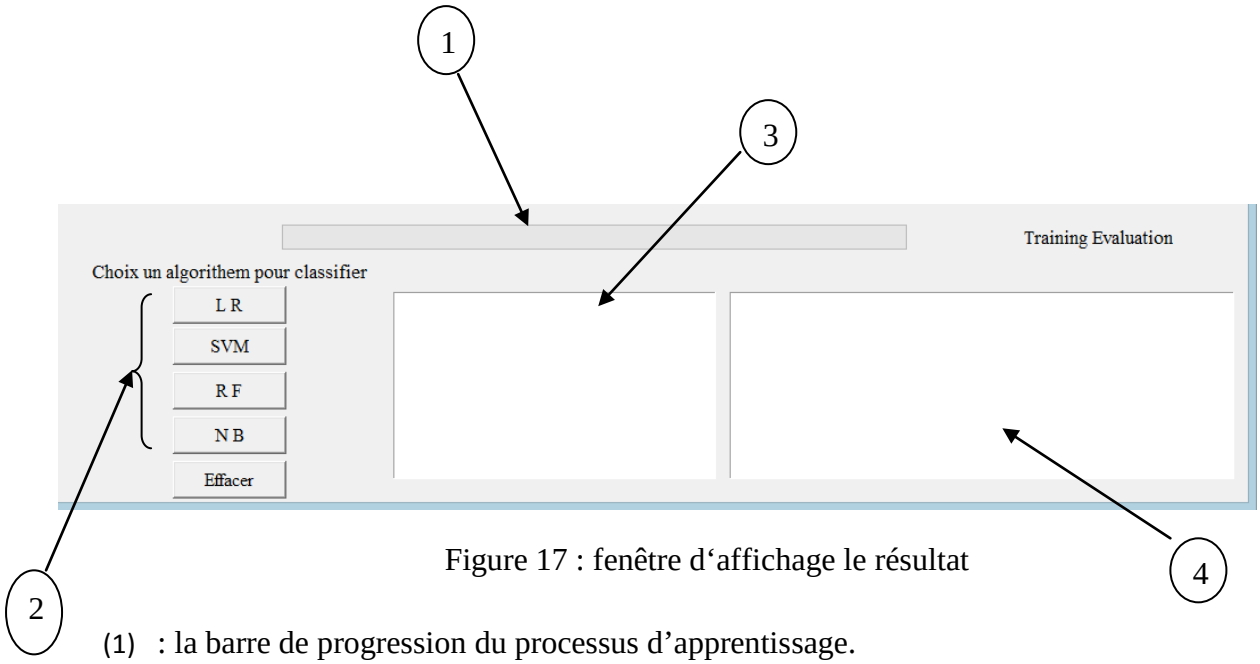

Figure 17 : fenêtre d'affichage le résultat

(1) : la barre de progression du processus d'apprentissage.

(2) : Les algorithmes de classification

(3) : Affichage du résultat de la classification

(4) : Affichage de la F-mesure

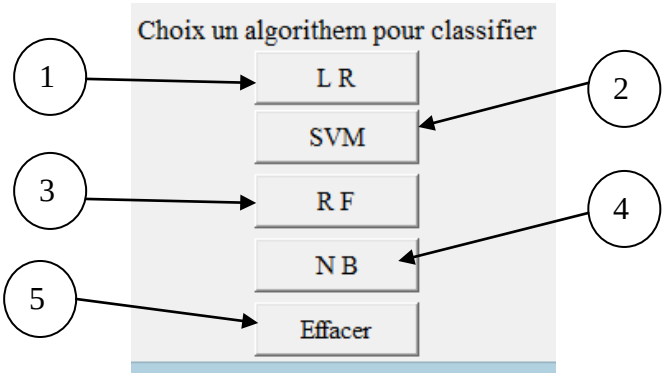

Figure 18 : fenêtre de choix de l'algorithme

(1) : L'algorithme de régression logistique (LR)

(2) : L'algorithme de Machine à vecteurs de support (SVM)

(3) : L'algorithme de Forêts d'arbres décisionnels (Random Forest)

(4) : L'algorithme de naïve bayésienne (NB)

(5) : Effacer tous les résultats

4. Exemple d'utilisation

Nous avons choisi un fichier texte, alors nous avons eu la figure suivante :

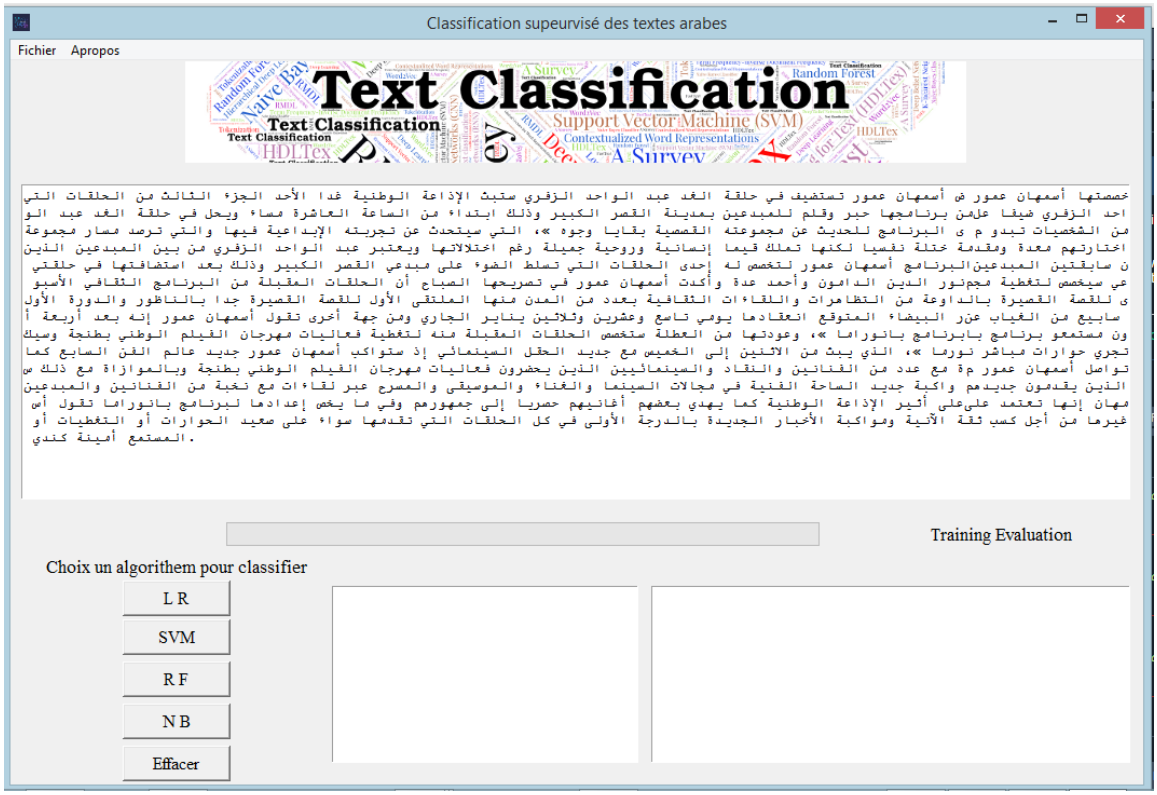

Figure 19 : Exemple d'introduire un fichier texte

On clique sur le « LR » pour afficher le résultat de classification de l'algorithme régression logistique :

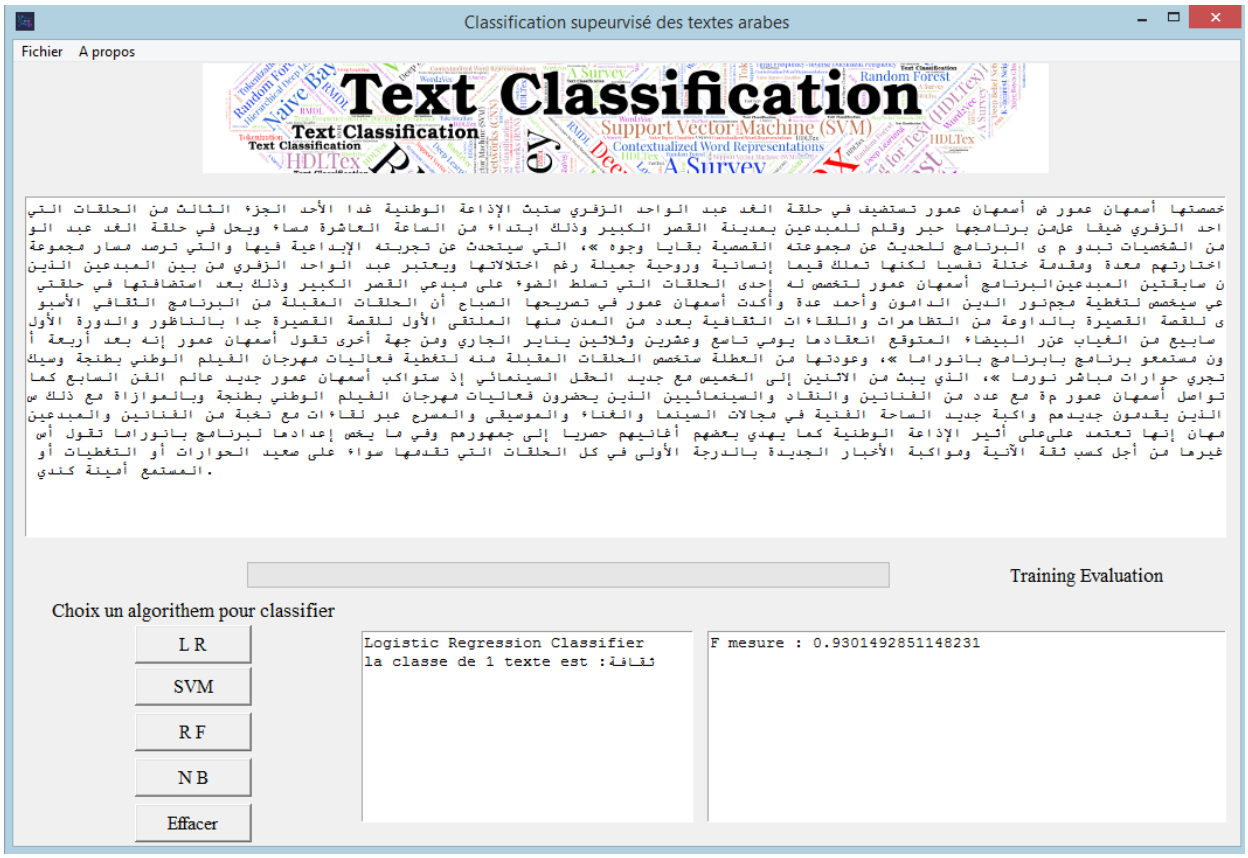

Figure 20 : Exemple de l'algorithme LR

5. Expérimentation et discussion

Après avoir choisi un texte, et sélectionné un algorithme de classification avec la méthode d'extraction des caractéristiques (features extraction), notre système déclenche un processus d'apprentissage sur une partie dite training-set représentant 70% d'un dataset. Ensuite, une phase d'évaluation consiste classifier le reste du dataset dit test-set et voir le taux de réussite du processus de classification.

5.1. Dataset

Dataset est une collection de textes arabes, qui couvre la langue arabe moderne utilisée dans les articles de journaux. Le texte contient des mots alphabétiques, numériques et symboliques. L'existence de mots numériques et symboliques dans cet ensemble de données pourrait indiquer l'efficacité et la robustesse de nombreux documents de classification et d'indexation de textes arabes.

Dataset se compose de 111 728 documents et de 319 254 124 mots structurés en fichiers texte, et collectés à partir de 3 journaux en ligne arabes : Assabah, Hespress et Akhbarona à l'aide d'un processus d'exploration Web semi-automatique. Les documents de l'ensemble de données sont classés en 5 classes : sport, politique, culture, économie et divers. Le nombre de documents et de mots pour chaque classe varie d'une classe à l'autre (Mohamed, 2018).

Ce dataset est trop gros pour être utilisé dans ce travail. Alors, nous avons pris 500 documents, car il existe une étude (Madhfar & Al-Hagery, 2019) qui dit que le nombre supérieur de 5000 documents, l'évaluation est reste stable (ne change pas).

5.2. Features Extraction

1- Bag Of Word : Cette méthode consiste à représenter le document sous forme d'un vecteur de mots. Le processus qui permet de convertir le texte d'un document à un ensemble de termes est appelé l'analyse lexicale qui permet de reconnaître les espaces de séparation des mots, les ponctuations, les chiffres, etc., pour qu'ils seront tous supprimés de la représentation. Cette représentation a comme avantage d'exclure toute analyse grammaticale et toute notion de distance entre les mots, mais présente comme inconvénient la difficulté de délimiter les mots dans certaines langues telles que l'Arabe ou l'Allemand (DERDRA et BENSFIA, 2012).

2- Term Frequency-Inverse Document Frequency (TF-IDF) :

- **TF (Term Frequency) :** La fréquence d'un terme est simplement le nombre d'occurrences de ce terme dans le document considéré.
- **IDF (Inverse Document Frequency) :** La fréquence inverse de document est une mesure de l'importance du terme dans l'ensemble du corpus.
- **TF*IDF:** Le poids d'un terme *T* dans un document *D* est calculé comme suit :

$$\boldsymbol{TFIDF\ (Ti, Dj)\ =\ TF\ (Ti, Dj)\ *\ log\ (N/\ DF(T))}$$

Avec :

- **TF (Ti, Dj)** : la fréquence du terme dans le document.
- **N** : le nombre total de documents de la base documentaire.
- **DF(Ti)** : le nombre de documents contenant le terme (DERDRA et BENSFIA, 2012).

5.3. Les critères d'évaluations utilisées

Ce sont les critères pour effectuer la comparaison entre les Classifieurs (Hocine, 2011) :

- **Compréhensibilité** : montre si le modèle est compréhensible et si le système donne des réponses qui permettent de comprendre pourquoi le document est classé dans une certaine classe.
- **Simplicité** : apprécie le taux de simplicité des résultats d'apprentissage produits par le classifieur.
- **Intelligibilité** : évalue le degré d'intelligence du classifieur.
- **Le temps de réponse et d'indexation** : est aussi un point qui peut être fondamental.
- L'encombrement **du système et les ressources en mémoire requises** : l'espace alloué en mémoire vive et sur le disque dur qui doit être prise en compte dans de nombreux cas.

Pour mesurer l'efficacité d'un classificateur dans un problème à n classes (en l'occurrence deux : positif et négatif), trois mesures sont utilisées : la précision, le rappel et le F-score**.**

Notations	Descriptions
TP	vrai positif
PF	faux positif
TN	vrai négatif
FN	faux négatif

Tableau 3 : les notions utilisées dans F-score.

1- Précision

Précision. Proportion d'éléments bien classés pour une classe donnée:

$$Precision = \frac{TP}{TP + FP}$$

2- Rappel

Proportion d'éléments bien classés par rapport au nombre d'éléments de la classe à prédire :

$$Rappel = \frac{TP}{TP + FN}$$

3- F-score

La mesure la plus utilisée en classification d'opinions c'est le F-score. Elle est calculée par la formule suivante :

$$F = \frac{2 \times \text{Precision} \times \text{Rappel}}{Precision + Rappel}$$

6. Etude comparative et discussion

Notre corpus contient 500 textes répartis en 5 catégories :

- Classe 0 c'est ثقافة (culture)

- Classe 1 c'est متنوعة (divers)
- Classe 2 c'est اقتصاد (économie)
- Classe 3 c'est سياسة (politique)
- Classe 4 c'est رياضة (sport)

Catégorie	Nombre de document
ثقافة	100
متنوعة	100
اقتصاد	100
سياسة	100
رياضة	100

Tableau 4 : Corpus d'apprentissage utilisé dans les expérimentations

Notre travail en utilise différent algorithme et différent Features Extraction, de chaque algorithme en utilise Features Extraction différent :

Algorithme	Features Extraction
LR	TFIDF
SVM	BOW
RF	Vecture de hachage
NB	TFIDF et BOW

Tableau 5 : Algorithme et Features Extraction

6.1. Précision, Rappel et F-score

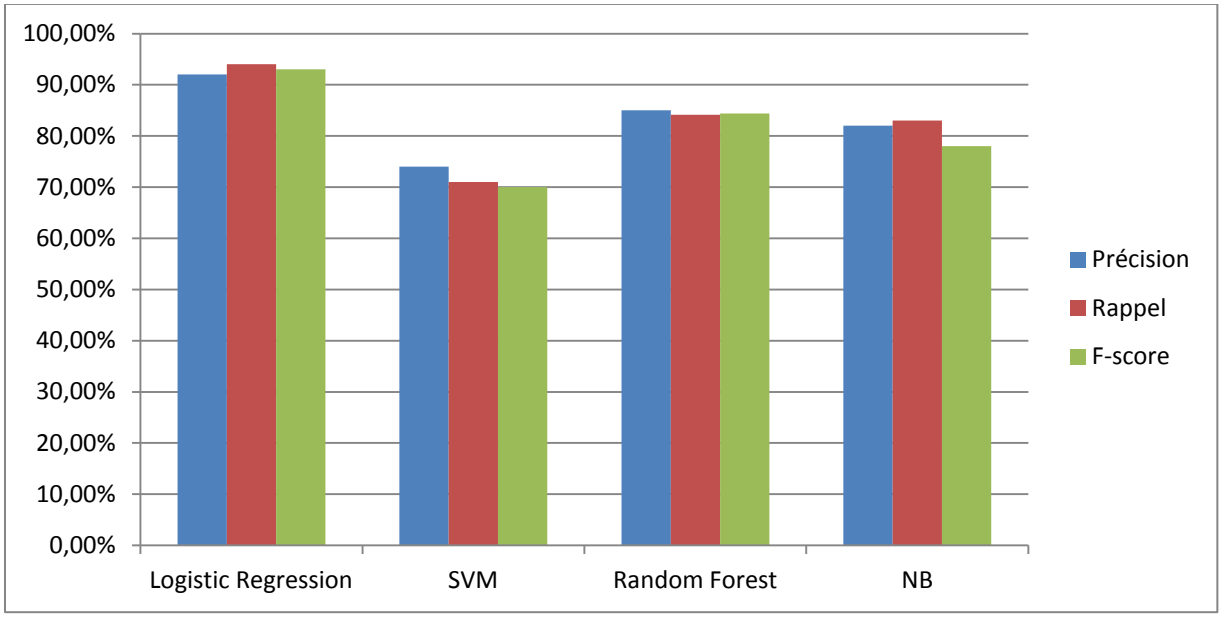

Figure 21 : Précision, Rappel et F-score

Selon les résultats obtenus (Figure 21), il est clair que la performance de l'approche Logistic Regression est meilleure que les autres approches par une exactitude de 93%, puis l'approche Random Forest par une exactitude de 84.39%, puis l'approche NB par une exactitude de 78% et enfin l'approche SVM par une exactitude de 70%.

7. Conclusion

Dans ce chapitre, nous avons présenté les interfaces et un guide d'utilisation de notre logiciel. Dans ce suit, une conclusion générale synthétise notre travail et trace quelques perspectives.

Conclusion générale et Perspectives

Dans ce mémoire de fin d'études, nous nous sommes intéressés à la classification supervisée (classement automatique ou encore catégorisation) des documents avec les méthodes LR, SVM, RF et NB. Rappelons que le but de la catégorisation est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu.

Cette phase d'apprentissage a pour but de catégoriser un corpus qui contient des documents de thèmes variés et d'associer à chaque document la catégorie adéquate.

Dans ce manuscrit, nous avons commencé par la définition de quelques concepts nécessaires pour comprendre notre projet de fin d'études. Nous avons présenté également les différentes techniques que nous avons utilisées dans nos expérimentations. Enfin, nous avons fourni une description détaillée de l'application qui applique plusieurs techniques de classification de texte. Après l'analyse des résultats obtenus, on a pu constater que les taux de classification sont acceptables.

Perspectives

Malheureusement, le temps attribué à ce travail était très court, d'où il était difficile de fixer certains paramètres pour étudier d'autres approches et algorithmes. Nous proposons comme perspectives :

- Appliquer d'autres approches de représentation des textes, à savoir : l'approche conceptuelle et l'approche des n-grammes.
- Implémenter d'autres classifieurs pour avoir l'occasion de les comparer avec notre classifieur.
- Utiliser les ontologies ou les dictionnaires pour enrichir la représentation des textes.
- Passer du Machine Learning au Deep Learning

Bibliographie et Webographie

- Adel Hamdan M. (2019). "Arabic Text Classification: A Review". Modern Applied Science; Vol. 13, No. 5; 2019.
- Ahlam W., Said A. S., Khaled S. (2021). "Text Classification of Arabic Text: Deep Learning in ANLP". In Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2021 (pp. 95-103). Springer International Publishing.
- Ahmed Z., Rabah M. (2019), "Catégorisation automatique des textes arabes". Université Blida, Octobre 2019.
- Andy L. (2012). "Documentation for R package randomForest" [archive], 16 octobre 2012.
- Ankit P., Muru S., Malaikannan S. (2020). "MAGNET: Multi-Label Text Classification using Attentio,n-based Graph Neural Network". In Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020), 2020.
- Billal B. (2017). "Classification supervisée des textes courts et Bruttés : application au domaine des médias sociaux". Université du QUEBAC à MONTREAL. Avril 2017.
- Caruana, R. and Niculescu-Mizil, A.(2006). "An empirical comparison of supervised learning algorithms". Proceedings of the 23rd international conference on Machine learning, 2006.
- Chen Y., Balke W-T., Xu J., Xu W., Jin P., Lin X., Tang T. et Hwang E. (2014). "Web-Age Infonnation Management". WAIM Intemational Workshops: BigEM, HardBD, DaNoS, HRSUNE, BIDASYS, Macau, China, June 16-18, 2014, Revised Selected Papers, volume 8597. Springer. 2014.
- Choayb O. (2014). «Classification automatique de textes ». UNIVERSITE DE M'SILA, 2014.
- DERDRA AMEL T., BENSFIA F. (2012). "La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue". Mémoire de Master, Université Abou Bakr Belkaid– Tlemcen, 2011-2012.
- Hanane T. (2018). "Classification automatique de textes". UNIVERSITE MOHAMED BOUDIAF - M' SILA. 2018.
- Harry Z. (2004). "The Optimality of Naive Bayes". Conférence FLAIRS. 2004.

- Ho, Tin K. (1995). "Random Decision Forests". Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14-16 august 1995, p. 278-282 .
- Hocine M. (2011). « classification Automatique de Textes Approche Orientée Agent », Magister en informatique, Février 2011.
- Imad Z., Anoual El K. (2021). "The effects of Pre-Processing Techniques on Arabic Text Classification". International Journal of Advanced Trends in Computer Science and Engineering, Volume 10, No.1, January - February 2021.
- Leo B. (2001). "Random Forests". Machine Learning, vol. 45, no 1, p. 5–32 (DOI 10.1023/A:1010933404324) , 2001.
- M. Hocine (2011). "Classification Automatique de Textes Approche Orientée Agent". Magister en informatique, Février 2011.
- Madhfar M. A. H., & Al-Hagery M. A. H. (2019, April). "Arabic text classification: A comparative approach using a big dataset". In 2019 International Conference on Computer and Information Sciences (ICCIS) (pp. 1-5). IEEE.
- Maurice R. (2006). "ALGORITHMES DE CLASSIFICATION". Université Paul Cézanne Marseille, France, Juin 2006.
- Mohamed Amine C., Djilali C. (2014). "Conception et Réalisation d'un lemmatiseur hybride de texte arabe", Université Ahmed Draya Adrar Algérie, 2014.
- Mohamed B. (2018). "DataSet for Arabic Classification". Mendeley Data, V2, doi: 10.17632/v524p5dhpj.2, 2018.
- Noureddine D. (2017). "Extraction de connaissances à partir du Texte". Université Djillali Liabes de Sidi Bel Abbes. 2016 – 2017.
- Raheel S. (2010). "L'Apprentissage Artificiel pour la Fouille de Données Multilingues: Application à la Classification Automatique des Documents Arabes". Thèse de doctorat en Sciences de l'Information et de la Communication, Université Lumière Lyon 2, 2010.
- Rasha E., Mahmoud Ali (2017). "Arabic Text Classification Process". International Journal of Computer Science and Software Engineering (IJCSSE), Volume 6, Issue 11, November 2017.
- Robert N., John E., Gary M. (2009). "Handbook for Statistical Analysis and Data Mining". Academic Press, Page 247 Edition 2009.

- SAHRAOUI S. (2013). "Identification de la langue et catégorisation thématique de textes d'un corpus multilingue en utilisant les réseaux de neurones artificiels RNA". Mémoire de Master, Université Mohamed BOUDIAF– Msila, Algérie, 2012-2013.
- Samir B., Mohamed B., Fatiha El A., Loubna Ch., Abd Elmajid El M.(2018). "Arabic Text Classification Using Deep Learning Technics". International Journal of Grid and Distributed Computing Vol. 11, No. 9 (2018), pp.103-114, 2018.
- Santiago G.C., Eduardo C. G.M. (2020). "Comparing BERT against traditional machine learning text classification". Madrid, Spain, 2020.
- Sebastian B., Marc H., Matthias B., Maximilien K.(2021). "Evaluation of Representation Models for Text Classification with AutoML Tools". Stuttgart, Germany, 2021.
- Sebastianni F. (2002). "Machine learning in automated text categorization". ACM Computing Surveys, 34(1) :1‑47, 2002.
- Shammur A C., Ahmed A., Kareem D., Soon-gyo J., Joni S., Bernard J J. (2020). "Improving Arabic Text Categorization Using Transformer Training Diversification". Proceedings of the Fifth Arabic Natural Language Processing Workshop, pages 226–236 Barcelona, Spain (Online), December 12, 2020.
- Shervin M., Nal K., Erik C., Narjes N., Meysam C., Jianfeng G.(2020). "Deep learning based text classification: A comprehensive review". CoRR, vol. abs/2004.03705, 2020.
- Simon R. (2005). "Catégorisation automatique de textes et Cooccurrence de mots provenant de documents non étiquetés". Mémoire, Université Laval Québec, Canada, Janvier 2005.
- Simon R. (2011). "Catégorisation automatique de textes et cooccurrence de mots : Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquètes". Editions universitaires européennes EUE, 2011.
- Toussaint Y. (2004). "Extraction de connaissances à partir de textes structures". Document Numérique, 8(3) :11–34, 2004.
- WISSAME A., NASSIMA C. (2020). "Classification des textes arabes". Mémoire de master, UNIVERSITE MOHAMED BOUDIAF - M'SILA, 2020.
- Xiaoyu L. (2021). "Efficient English text classification using selected Machine Learning Techniques". Alexandria Engineering Journal 21 February 2021 60, 3401–3409

- Yujia B., Menghua W., Shiyu C., and Regina B. (2020). "Few-shot text classification with distributional signatures". in Proc. ICLR, 2020, 2020.
- Guillaume S.C. (2019). (https://machinelearnia.com/comment-fonctionne-machine-learning/), 2021. (Vu le 10/02/2021)
- Younes B. (2016). (https://mrmint.fr/introduction-machine-learning), 2016 – 2017. (Vu le 10/02/2021)
- Younes B. (2018). (https://mrmint.fr/logistic-regression-machine-learning-introduction-simple), 2016 – 2017. (Vu le 10/02/2021)
- Site 1 : (https://evalorix.com/boutique/produits-et-outils-en-sciences-et-genie/produits-et-outils-en-informatique/le-taln-c-est-quoi-au-juste/), (**eValorix,** Caroline Barrière), 2021. (Vu le 11/02/2021)
- Site 2 : (https://whatis.techtarget.com/fr/definition/Machine-Learning?_gl=1*1rc988h*_ga*NDM5MTc0MTA0LjE1OTkyMjA0NjU.*_ga_RRBYR9CGB9*MTYxMjY5Mjg2MS4xLjAuMTYxMjY5Mjg2MS4w&_ga=2.191089004.714383947.1612692864-439174104.1599220465), (TechTarget, 2021). (Vu le 11/02/2021)
- Site 3: (https://dataanalyticspost.com/Lexique/random-forest/), (Data Analytics Post). (Vu le 25/06/2021)